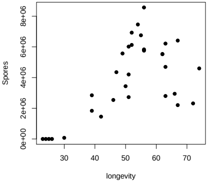
Nonlinear Models

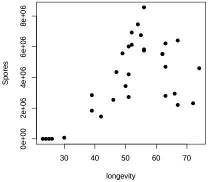
What do you do when you don't have a line?



What do you do when you don't have a line?

1. If nonlinear terms are additive fit with OLS
2. Transform? But think about what it will do to error.
3. Nonlinear Least Squares
4. Generalized Linear Models

A Quadratic Adventure



$$Spores = b0 + b1 * Longevity + b2 * Longevity^2 + error$$

Putting Nonlinear Terms into an Additive Model

```r
fungus.lmsq <- lm(Spores ~  longevity +  I(longevity^2), data=fungus)
```

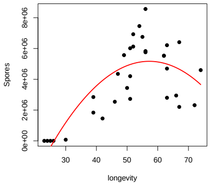
Parameters are the Same as Ever

```r
summary(fungus.lmsq)

#
# Call:
# lm(formula = Spores ~ longevity + I(longevity^2), data = fungus)
#
# Residuals:
#      Min       1Q   Median       3Q      Max
# -2467932 -1474558   444878  1068889  3407021
#
# Coefficients:
#                 Estimate Std. Error t value Pr(>|t|)
# (Intercept)    -12432138    2883786   -4.31  0.00017
# longevity         615120     126837    4.85  3.9e-05
# I(longevity^2)     -5374       1328   -4.05  0.00035
#
# Residual standard error: 1590000 on 29 degrees of freedom
# Multiple R-squared: 0.607, Adjusted R-squared: 0.58
# F-statistic: 22.4 on 2 and 29 DF, p-value: 1.3e-06
```

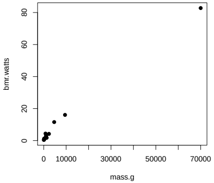
We Can't use abline Anymore

```r
plot(Spores ~ longevity, data=fungus, pch=19)
#
fungusFun <- function(x) coef(fungus.lmsq)[1] +
  coef(fungus.lmsq)[2]*x +
  coef(fungus.lmsq)[3]*x^2
#
curve(fungusFun, add=T, col="red", lwd=2)
```

We Can't use abline Anymore



What if It's not a Linear Combination of Terms?



$$MetabolicRate = a * mass^b$$

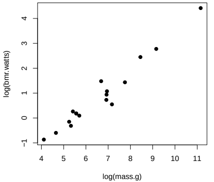
Common Transformations

- log(y)
- arcsin(sqrt(y)) for bounded data
- logit for bounded data (more well behaved)
- Box-Cox Transform

May have to add 0.01, 0.5, or 1 to many of these in cases with 0s

You must ask yourself, what do the transformed variables mean?

But Where does Error Come In



$$log(MetabolicRate) = log(a) + b * log(mass) + error$$

But Where does Error Come In

$$log(MetabolicRate) = log(a) + b * log(mass) + error$$

implies

$$MetabolicRate = a * mass^b * e^{error}$$

but we often want

$$MetabolicRate = a * mass^b + error$$

Nonlinear Least Squares Fitting

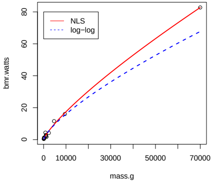
```r
primate.nls <- nls(bmr.watts ~ a*mass.g^b, data=primates,
                   start=list(a = 0.0172858, b = 0.74160))
```

Uses algorithm for fitting. Very flexible. Must specify start values.

Nonlinear Least Squares Fitting

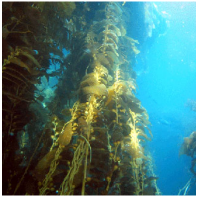
```
summary(primate.nls)

#
# Formula: bmr.watts ~ a * mass.g^b
#
# Parameters:
#   Estimate Std. Error t value Pr(>|t|)
# a  0.01106    0.00225    4.93  0.00018
# b  0.79956    0.01842   43.41  < 2e-16
#
# Residual standard error: 0.982 on 15 degrees of freedom
#
# Number of iterations to convergence: 4
# Achieved convergence tolerance: 7.01e-07
```

NLS Performs Better



Exercise: Kelp!

- Evaluate the Frond ∼ Holdfast relationship
- Fit a model with a log transformation
- Fit a model with a nls model
- Compare
- Check the diagnostics - see anything?

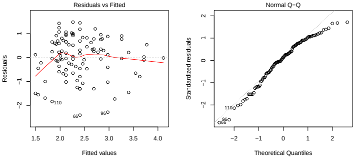


The Kelp Data

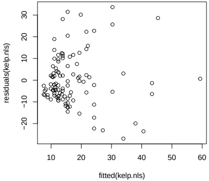
```
# Error:  object 'kelp' not found
```

FRONDS are a count variable, cannot be < 0

Envelope Residuals from Log Transform

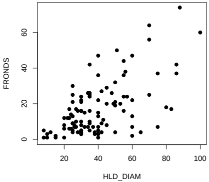


Mild Trumpet even in NLS



Maybe the error is wrong...

The Kelp Data



FRONDS are a count variable, cannot be < 0

Generalized Linear Models

Basic Premise: $y \sim dist(\eta, \nu)$

dist is a distribution of the exponential family
η is a link function such that $\eta = f(\mu)$ where μ is the mean of a curve
ϕ is a variance function

For example, if dist is Normal, canonical link is μ, variance is σ^2

Some Common Links

- Identity: $\eta = \mu$ - e.g. $\mu = a + bx$
- Log: $\eta = log(\mu)$ - e.g. $\mu = e^(a + bx)$
- Logit: $\eta = logit(\mu)$ - e.g. $\mu = \frac{e^(a+bx)}{1+e^(a+bx)}$
- Inverse: $\eta = \frac{1}{\mu}$ - e.g. $\mu = (a + bx)^{-1}$

Distributions, Canonical Links, and Dispersion

Distribution	Canonical Link	Variance Function
Normal	identity	1
Poisson	log	μ
Quasipoisson	log	$\mu\theta$
Binomial	logit	$\mu(1-\mu)$
Quasibinomial	logit	$\mu(1-\mu)\theta$
Negative Binomial	log	$\mu + \kappa\mu^2$
Gamma	inverse	μ^2
Inverse Normal	$1/\mu^2$	μ^3

Distributions and Other Links

Distribution	Links
Normal	identity, log, inverse
Poisson	log, identity, sqrt
Quasipoisson	log, identity, sqrt
Binomial	logit, probit, cauchit, log, log-log
Quasibinomial	logit, probit, cauchit, log, log-log
Negative Binomial	log, identity, sqrt
Gamma	inverse, identity, log
Inverse Normal	$1/\mu^2$, inverse, identity, log

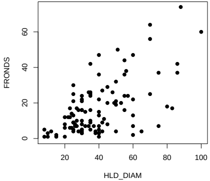
Deviance and IWLS

Every GLM has a Set of Deviance Function to be Minimized

i.e., for a normal distribution $D_M = \sum(y_i - \hat{\mu}_i)^2$

Models are Fit using Iteratively Weighted Least Squares algorithm

The Kelp Data



FRONDS are a count variable, cannot be < 0

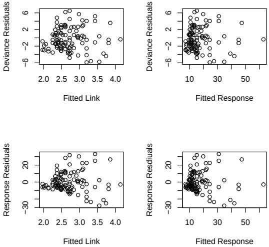
Fitting a GLM with a Poisson Error and Log Link

Fronds $\sim$ Poisson($\hat{Fronds}$)

$\hat{Fronds} =$ exp(a + b * holdfast diameter)

```
kelp.glm <- glm(FRONDS ~ HLD_DIAM, data=kelp,
                family=poisson(link="log"))
```

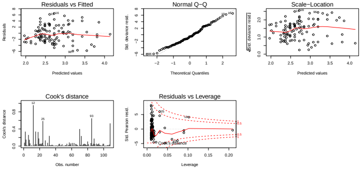
How do we Assess Meeting Assumptions?



Different Types of Residuals

```
residuals(kelp.glm, type="deviance")
residuals(kelp.glm, type="pearson")
residuals(kelp.glm, type="response")
```

How do we Assess Meeting Assumptions?



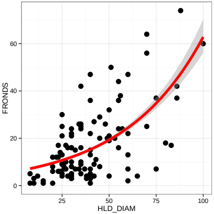
GLM Model Coefficients

```
#
# Call:
# glm(formula = FRONDS ~ HLD_DIAM, family = poisson(link = "log"),
#     data = kelp)
#
# Deviance Residuals:
#    Min      1Q  Median      3Q     Max
# -5.902  -2.387  -0.557   1.613   6.512
#
# Coefficients:
#             Estimate Std. Error z value Pr(>|z|)
# (Intercept)  1.77806    0.05726    31.1   <2e-16
# HLD_DIAM     0.02362    0.00105    22.5   <2e-16
#
# (Dispersion parameter for poisson family taken to be 1)
#
#     Null deviance: 1289.17  on 107  degrees of freedom
# Residual deviance:  832.56  on 106  degrees of freedom
#   (32 observations deleted due to missingness)
```

Checking Fit

```
cor(fitted(kelp.glm),
    fitted(kelp.glm) + residuals(kelp.glm, type="response"))^2

# [1] 0.3649

summary(kelp.lm)$r.squared

# [1] 0.277
```

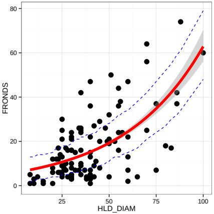
The Fitted Model



Prediction Confidence Intervals by Hand

```
upperCI <- qpois(0.975, lambda = round(fitted(kelp.glm)))
lowerCI <-  qpois(0.025, lambda = round(fitted(kelp.glm)))
HLD <- na.omit(kelp)$HLD_DIAM
#
kelp.ggplot +
  geom_line(mapping=aes(x=HLD,  y=upperCI), lty=2, col="blue") +
  geom_line(mapping=aes(x=HLD,  y=lowerCI), lty=2, col="blue")
```

Prediction Confidence Intervals by Hand



Overdispersion?

Which Overdispersed Distribution to Use?

v(quasipoisson) $= \mu\theta$

v(Negative Binomial) $= \mu + \kappa\mu^2$

see Ver Hoef and Boveng 2007 Ecology

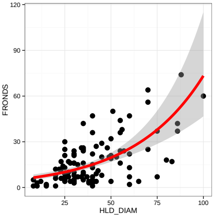
GLM with Negative Binomial

```r
library(MASS)
#
kelp.glm.nb <- glm.nb(FRONDS ~ HLD_DIAM, data=kelp)
```

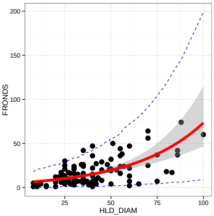
Negative Binomial Performs Better

```r
anova(kelp.glm, kelp.glm.nb)

# Analysis of Deviance Table
#
# Model 1: FRONDS ~ HLD_DIAM
# Model 2: FRONDS ~ HLD_DIAM
#   Resid. Df Resid. Dev Df Deviance
# 1       106        833
# 2       106        114  0      718
```

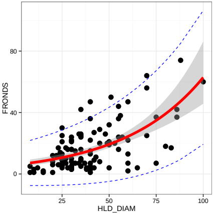
The Fitted Model



Fit with Prediction Error



Compare to Quasipoisson with Prediction Error



Example: Wolf Inbreeding and Litter Size

- The Number of Pups is a Count!
- Fit GLMs with different errors and links
- Which is the best model?
- Plot with fit and prediction error